Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'mailx.1p' command

$ man mailx.1p

MAILX(1P) POSIX Programmer's Manual MAILX(1P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

mailx ? process messages

SYNOPSIS

Send Mode

mailx [-s subject] address...

Receive Mode

mailx -e

mailx [-HiNn] [-F] [-u user]

mailx -f [-HiNn] [-F] [file]

DESCRIPTION

The mailx utility provides a message sending and receiving facility. It has two major modes, selected by the options used: Send Mode and Re? ceive Mode.

On systems that do not support the User Portability Utilities option, an application using mailx shall have the ability to send messages in an unspecified manner (Send Mode). Unless the first character of one or more lines is <tilde> ('~'), all characters in the input message shall

appear in the delivered message, but additional characters may be in?
serted in the message before it is retrieved.

On systems supporting the User Portability Utilities option, mail-re?
ceiving capabilities and other interactive features, Receive Mode, de?
scribed below, also shall be enabled.

Send Mode

Send Mode can be used by applications or users to send messages from the text in standard input.

Receive Mode

Receive Mode is more oriented towards interactive users. Mail can be read and sent in this interactive mode.

When reading mail, mailx provides commands to facilitate saving, delet?
ing, and responding to messages. When sending mail, mailx allows edit?
ing, reviewing, and other modification of the message as it is entered.

Incoming mail shall be stored in one or more unspecified locations for each user, collectively called the system mailbox for that user. When mailx is invoked in Receive Mode, the system mailbox shall be the de?
fault place to find new mail. As messages are read, they shall be marked to be moved to a secondary file for storage, unless specific ac?
tion is taken. This secondary file is called the mbox and is normally located in the directory referred to by the HOME environment variable (see MBOX in the ENVIRONMENT VARIABLES section for a description of this file). Messages shall remain in this file until explicitly re?
moved. When the -f option is used to read mail messages from secondary files, messages shall be retained in those files unless specifically removed. All three of these locations?system mailbox, mbox, and sec?
ondary file?are referred to in this section as simply ``mailboxes'', unless more specific identification is required.

OPTIONS

The mailx utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines.

The following options shall be supported. (Only the -s subject option shall be required on all systems. The other options are required only

on systems supporting the User Portability Utilities option.)

-e Test for the presence of mail in the system mailbox. The mailx utility shall write nothing and exit with a successful return code if there is mail to read.

-f Read messages from the file named by the file operand instead of the system mailbox. (See also folder.) If no file operand is specified, read messages from mbox instead of the system mailbox.

-F Record the message in a file named after the first recipient. The name is the login-name portion of the address found first on the To: line in the mail header. Overrides the record variable, if set (see Internal Variables in mailx).

-H Write a header summary only.

-i Ignore interrupts. (See also ignore.)

-n Do not initialize from the system default start-up file. See the EXTENDED DESCRIPTION section.

-N Do not write an initial header summary.

-s subject

Set the Subject header field to subject. All characters in the subject string shall appear in the delivered message. The results are unspecified if subject is longer than {LINE_MAX} - 10 bytes or contains a <newline>.

-u user Read the system mailbox of the login name user. This shall only be successful if the invoking user has appropriate priv? ileges to read the system mailbox of that user.

OPERANDS

The following operands shall be supported:

address Addressee of message. When -n is specified and no user start-up files are accessed (see the EXTENDED DESCRIPTION section), the user or application shall ensure this is an address to pass to the mail delivery system. Any system or user start-up files may enable aliases (see alias under Commands in mailx) that may modify the form of address before it is passed to

the mail delivery system.

file A pathname of a file to be read instead of the system mailbox when -f is specified. The meaning of the file option-argument shall be affected by the contents of the folder internal variable; see Internal Variables in mailx.

STDIN

When mailx is invoked in Send Mode (the first synopsis line), standard input shall be the message to be delivered to the specified addresses. When in Receive Mode, user commands shall be accepted from stdin. If the User Portability Utilities option is not supported, standard input lines beginning with a <tilde> ('~') character produce unspecified re? sults.

If the User Portability Utilities option is supported, then in both Send and Receive Modes, standard input lines beginning with the escape character (usually <tilde> ('~')) shall affect processing as described in Command Escapes in mailx.

INPUT FILES

When mailx is used as described by this volume of POSIX.1?2017, the file option-argument (see the -f option) and the mbox shall be text files containing mail messages, formatted as described in the OUTPUT FILES section. The nature of the system mailbox is unspecified; it need not be a file.

ENVIRONMENT VARIABLES

Some of the functionality described in this section shall be provided on implementations that support the User Portability Utilities option as described in the text, and is not further shaded for this option.

The following environment variables shall affect the execution of mailx:

DEAD Determine the pathname of the file in which to save partial messages in case of interrupts or delivery errors. The de? fault shall be dead.letter in the directory named by the HOME variable. The behavior of mailx in saving partial messages is unspecified if the User Portability Utilities option is not

supported and DEAD is not defined with the value /dev/null.

EDITOR Determine the name of a utility to invoke when the edit (see Commands in mailx) or ~e (see Command Escapes in mailx) com? mand is used. The default editor is unspecified. On XSI-con? formant systems it is ed. The effects of this variable are unspecified if the User Portability Utilities option is not supported.

HOME Determine the pathname of the user's home directory.

LANG Provide a default value for the internationalization vari? ables that are unset or null. (See the Base Definitions vol? ume of POSIX.1?2017, Section 8.2, Internationalization Vari? ables for the precedence of internationalization variables used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and the handling of case-insensitive address and header-field comparisons.

LC_TIME This variable may determine the format and contents of the date and time strings written by mailx. This volume of POSIX.1?2017 specifies the effects of this variable only for systems supporting the User Portability Utilities option.

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

LISTER Determine a string representing the command for writing the contents of the folder directory to standard output when the folders command is given (see folders in Commands in mailx). Any string acceptable as a command_string operand to the sh -c command shall be valid. If this variable is null or not

set, the output command shall be ls. The effects of this variable are unspecified if the User Portability Utilities option is not supported.

MAILRC Determine the pathname of the user start-up file. The default shall be .mailrc in the directory referred to by the HOME environment variable. The behavior of mailx is unspecified if the User Portability Utilities option is not supported and MAILRC is not defined with the value /dev/null.

MBOX Determine a pathname of the file to save messages from the system mailbox that have been read. The exit command shall override this function, as shall saving the message explicitly in another file. The default shall be mbox in the directory named by the HOME variable. The effects of this variable are unspecified if the User Portability Utilities option is not supported.

NLSPATH Determine the location of message catalogs for the processing of LC_MESSAGES.

PAGER Determine a string representing an output filtering or pagination command for writing the output to the terminal. Any string acceptable as a command_string operand to the sh -c command shall be valid. When standard output is a terminal device, the message output shall be piped through the command if the mailx internal variable crt is set to a value less the number of lines in the message; see Internal Variables in mailx. If the PAGER variable is null or not set, the paginator shall be either more or another paginator utility documented in the system documentation. The effects of this variable are unspecified if the User Portability Utilities option is not supported.

SHELL Determine the name of a preferred command interpreter. The default shall be sh. The effects of this variable are unspecified if the User Portability Utilities option is not supported.

TERM If the internal variable screen is not specified, determine the name of the terminal type to indicate in an unspecified manner the number of lines in a screenful of headers. If TERM is not set or is set to null, an unspecified default terminal type shall be used and the value of a screenful is unspecified. The effects of this variable are unspecified if the User Portability Utilities option is not supported.

TZ This variable may determine the timezone used to calculate date and time strings written by mailx. If TZ is unset or null, an unspecified default timezone shall be used.

VISUAL Determine a pathname of a utility to invoke when the visual command (see Commands in mailx) or ~v command-escape (see Command Escapes in mailx) is used. If this variable is null or not set, the full-screen editor shall be vi. The effects of this variable are unspecified if the User Portability Utilities option is not supported.

ASYNCHRONOUS EVENTS

When mailx is in Send Mode and standard input is not a terminal, it shall take the standard action for all signals.

In Receive Mode, or in Send Mode when standard input is a terminal, if a SIGINT signal is received:

1. If in command mode, the current command, if there is one, shall be aborted, and a command-mode prompt shall be written.
2. If in input mode:
   a. If ignore is set, mailx shall write "@\n", discard the current input line, and continue processing, bypassing the message-abort mechanism described in item 2b.
   b. If the interrupt was received while sending mail, either when in Receive Mode or in Send Mode, a message shall be written, and another subsequent interrupt, with no other intervening characters typed, shall be required to abort the mail message. If in Receive Mode and another interrupt is received, a com? mand-mode prompt shall be written. If in Send Mode and another

interrupt is received, mailx shall terminate with a non-zero status.

In both cases listed in item b, if the message is not empty:

i. If save is enabled and the file named by DEAD can be cre? ated, the message shall be written to the file named by DEAD. If the file exists, the message shall be written to replace the contents of the file.

ii. If save is not enabled, or the file named by DEAD cannot be created, the message shall not be saved.

The mailx utility shall take the standard action for all other signals.

STDOUT

In command and input modes, all output, including prompts and messages, shall be written to standard output.

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

Various mailx commands and command escapes can create or add to files, including the mbox, the dead-letter file, and secondary mailboxes. When mailx is used as described in this volume of POSIX.1?2017, these files shall be text files, formatted as follows:

line beginning with From<space>
[one or more header-lines; see Commands in mailx]
empty line
[zero or more body lines
empty line]
[line beginning with From<space>...]

where each message begins with the From <space> line shown, preceded by the beginning of the file or an empty line. (The From <space> line is considered to be part of the message header, but not one of the header-lines referred to in Commands in mailx; thus, it shall not be affected by the discard, ignore, or retain commands.) The formats of the remain? der of the From <space> line and any additional header lines are un? specified, except that none shall be empty. The format of a message

body line is also unspecified, except that no line following an empty line shall start with From <space>; mailx shall modify any such user-entered message body lines (following an empty line and beginning with From <space>) by adding one or more characters to precede the 'F'; it may add these characters to From <space> lines that are not preceded by an empty line.

When a message from the system mailbox or entered by the user is not a text file, it is implementation-defined how such a message is stored in files written by mailx.

EXTENDED DESCRIPTION

The functionality in the entire EXTENDED DESCRIPTION section shall be provided on implementations supporting the User Portability Utilities option. The functionality described in this section shall be provided on implementations that support the User Portability Utilities option (and the rest of this section is not further shaded for this option).

The mailx utility need not support for all character encodings in all circumstances. For example, inter-system mail may be restricted to 7-bit data by the underlying network, 8-bit data need not be portable to non-internationalized systems, and so on. Under these circumstances, it is recommended that only characters defined in the ISO/IEC 646:1991 standard International Reference Version (equivalent to ASCII) 7-bit range of characters be used.

When mailx is invoked using one of the Receive Mode synopsis forms, it shall write a page of header-summary lines (if -N was not specified and there are messages, see below), followed by a prompt indicating that mailx can accept regular commands (see Commands in mailx); this is termed command mode. The page of header-summary lines shall contain the first new message if there are new messages, or the first unread message if there are unread messages, or the first message. When mailx is invoked using the Send Mode synopsis and standard input is a termi? nal, if no subject is specified on the command line and the asksub variable is set, a prompt for the subject shall be written. At this point, mailx shall be in input mode. This input mode shall also be en?

tered when using one of the Receive Mode synopsis forms and a reply or new message is composed using the reply, Reply, followup, Followup, or mail commands and standard input is a terminal. When the message is typed and the end of the message is encountered, the message shall be passed to the mail delivery software. Commands can be entered by begin? ning a line with the escape character (by default, <tilde> ('~')) fol? lowed by a single command letter and optional arguments. See Commands in mailx for a summary of these commands. It is unspecified what effect these commands will have if standard input is not a terminal when a message is entered using either the Send Mode synopsis, or the Read Mode commands reply, Reply, followup, Followup, or mail.

Note: For notational convenience, this section uses the default es? cape character, <tilde>, in all references and examples.

At any time, the behavior of mailx shall be governed by a set of envi? ronmental and internal variables. These are flags and valued parame? ters that can be set and cleared via the mailx set and unset commands.

Regular commands are of the form:

[command] [msglist] [argument ...]

If no command is specified in command mode, next shall be assumed. In input mode, commands shall be recognized by the escape character, and lines not treated as commands shall be taken as input for the message.

In command mode, each message shall be assigned a sequential number, starting with 1.

All messages have a state that shall affect how they are displayed in the header summary and how they are retained or deleted upon termina? tion of mailx. There is at any time the notion of a current message, which shall be marked by a '>' at the beginning of a line in the header summary. When mailx is invoked using one of the Receive Mode synopsis forms, the current message shall be the first new message, if there is a new message, or the first unread message if there is an unread mes? sage, or the first message if there are any messages, or unspecified if there are no messages in the mailbox. Each command that takes an op? tional list of messages (msglist) or an optional single message (mes?

sage) on which to operate shall leave the current message set to the highest-numbered message of the messages specified, unless the command deletes messages, in which case the current message shall be set to the first undeleted message (that is, a message not in the deleted state) after the highest-numbered message deleted by the command, if one ex? ists, or the first undeleted message before the highest-numbered mes? sage deleted by the command, if one exists, or to an unspecified value if there are no remaining undeleted messages. All messages shall be in one of the following states:

new The message is present in the system mailbox and has not been viewed by the user or moved to any other state. Messages in state new when mailx quits shall be retained in the system mailbox.

unread The message has been present in the system mailbox for more than one invocation of mailx and has not been viewed by the user or moved to any other state. Messages in state unread when mailx quits shall be retained in the system mailbox.

read The message has been processed by one of the following com? mands: ~f, ~m, ~F, ~M, copy, mbox, next, pipe, print, Print, top, type, Type, undelete. The delete, dp, and dt commands may also cause the next message to be marked as read, depend? ing on the value of the autoprint variable. Messages that are in the system mailbox and in state read when mailx quits shall be saved in the mbox, unless the internal variable hold was set. Messages that are in the mbox or in a secondary mailbox and in state read when mailx quits shall be retained in their current location.

deleted The message has been processed by one of the following com? mands: delete, dp, dt. Messages in state deleted when mailx quits shall be deleted. Deleted messages shall be ignored un? til mailx quits or changes mailboxes or they are specified to the undelete command; for example, the message specification /string shall only search the subject lines of messages that

have not yet been deleted, unless the command operating on the list of messages is undelete. No deleted message or deleted message header shall be displayed by any mailx com? mand other than undelete.

preserved The message has been processed by a preserve command. When mailx quits, the message shall be retained in its current lo? cation.

saved The message has been processed by one of the following com? mands: save or write. If the current mailbox is the system mailbox, and the internal variable keepsave is set, messages in the state saved shall be saved to the file designated by the MBOX variable (see the ENVIRONMENT VARIABLES section). If the current mailbox is the system mailbox, messages in the state saved shall be deleted from the current mailbox, when the quit or file command is used to exit the current mailbox.

The header-summary line for each message shall indicate the state of the message.

Many commands take an optional list of messages (msglist) on which to operate, which defaults to the current message. A msglist is a list of message specifications separated by <blank> characters, which can in? clude:

n Message number n.

+ The next undeleted message, or the next deleted message for the undelete command.

- The next previous undeleted message, or the next previous deleted message for the undelete command.

. The current message.

^ The first undeleted message, or the first deleted message for the undelete command.

$ The last message.

* All messages.

n?m An inclusive range of message numbers.

address All messages from address; any address as shown in a header

summary shall be matchable in this form.

/string All messages with string in the subject line (case ignored).

:c All messages of type c, where c shall be one of:

d Deleted messages.

n New messages.

o Old messages (any not in state read or new).

r Read messages.

u Unread messages.

Other commands take an optional message (message) on which to operate, which defaults to the current message. All of the forms allowed for ms? glist are also allowed for message, but if more than one message is specified, only the first shall be operated on.

Other arguments are usually arbitrary strings whose usage depends on the command involved.

Start-Up in mailx

At start-up time, mailx shall take the following steps in sequence:

1. Establish all variables at their stated default values.
2. Process command line options, overriding corresponding default val? ues.
3. Import any of the DEAD, EDITOR, MBOX, LISTER, PAGER, SHELL, or VIS? UAL variables that are present in the environment, overriding the corresponding default values.
4. Read mailx commands from an unspecified system start-up file, un? less the -n option is given, to initialize any internal mailx vari? ables and aliases.
5. Process the user start-up file of mailx commands named in the user MAILRC variable.

Most regular mailx commands are valid inside start-up files, the most common use being to set up initial display options and alias lists. The following commands shall be invalid in a start-up file: !, edit, hold, mail, preserve, reply, Reply, shell, visual, Copy, followup, and Fol? lowup. Any errors in a start-up file shall either cause mailx to ter? minate with a diagnostic message and a non-zero status or to continue

after writing a diagnostic message, ignoring the remainder of the lines in the file.

A blank line in a start-up file shall be ignored.

Internal Variables in mailx

The following variables are internal mailx variables. Each internal variable can be set via the mailx set command at any time. The unset and set no name commands can be used to erase variables.

In the following list, variables shown as:

variable

represent Boolean values. Variables shown as:

variable=value

shall be assigned string or numeric values. For string values, the rules in Commands in mailx concerning filenames and quoting shall also apply.

The defaults specified here may be changed by the unspecified system start-up file unless the user specifies the -n option.

allnet All network names whose login name components match shall be treated as identical. This shall cause the msglist message specifications to behave similarly. The default shall be noallnet. See also the alternates command and the metoo variable.

append Append messages to the end of the mbox file upon termination instead of placing them at the beginning. The default shall be noappend. This variable shall not affect the save command when saving to mbox.

ask, asksub

Prompt for a subject line on outgoing mail if one is not specified on the command line with the -s option. The ask and asksub forms are synonyms; the system shall refer to asksub and noasksub in its messages, but shall accept ask and noask as user input to mean asksub and noasksub. It shall not be possible to set both ask and noasksub, or noask and asksub. The default shall be asksub, but no prompting shall be done

if standard input is not a terminal.

askbcc Prompt for the blind copy list. The default shall be noaskbcc.

askcc Prompt for the copy list. The default shall be noaskcc.

autoprint Enable automatic writing of messages after delete and un? delete commands. The default shall be noautoprint.

bang Enable the special-case treatment of <exclamation-mark> char? acters ('!') in escape command lines; see the escape command and Command Escapes in mailx. The default shall be nobang, disabling the expansion of '!' in the command argument to the ~! command and the ~<!command escape.

cmd=command

Set the default command to be invoked by the pipe command. The default shall be nocmd.

crt=number

Pipe messages having more than number lines through the com? mand specified by the value of the PAGER variable. The de? fault shall be nocrt. If it is set to null, the value used is implementation-defined.

debug Enable verbose diagnostics for debugging. Messages are not delivered. The default shall be nodebug.

dot When dot is set, a <period> on a line by itself during mes? sage input from a terminal shall also signify end-of-file (in addition to normal end-of-file). The default shall be nodot. If ignoreeof is set (see below), a setting of nodot shall be ignored and the <period> is the only method to terminate in? put mode.

escape=c Set the command escape character to be the character 'c'. By default, the command escape character shall be <tilde>. If escape is unset, <tilde> shall be used; if it is set to null, command escaping shall be disabled.

flipr Reverse the meanings of the R and r commands. The default shall be noflipr.

folder=directory

The default directory for saving mail files. User-specified filenames beginning with a <plus-sign> ('+') shall be ex? panded by preceding the filename with this directory name to obtain the real pathname. If directory does not start with a <slash> ('/'), the contents of HOME shall be prefixed to it. The default shall be nofolder. If folder is unset or set to null, user-specified filenames beginning with '+' shall refer to files in the current directory that begin with the literal '+' character. See also outfolder below. The folder value need not affect the processing of the files named in MBOX and DEAD.

header Enable writing of the header summary when entering mailx in Receive Mode. The default shall be header.

hold Preserve all messages that are read in the system mailbox in? stead of putting them in the mbox save file. The default shall be nohold.

ignore Ignore interrupts while entering messages. The default shall be noignore.

ignoreeof Ignore normal end-of-file during message input. Input can be terminated only by entering a <period> ('.') on a line by itself or by the ~. command escape. The default shall be noignoreeof. See also dot above.

indentprefix=string

A string that shall be added as a prefix to each line that is inserted into the message by the ~m command escape. This variable shall default to one <tab>.

keep When a system mailbox, secondary mailbox, or mbox is empty, truncate it to zero length instead of removing it. The de? fault shall be nokeep.

keepsave Keep the messages that have been saved from the system mail? box into other files in the file designated by the variable MBOX, instead of deleting them. The default shall be nokeep?

save.

metoo Suppress the deletion of the login name of the user from the recipient list when replying to a message or sending to a group. The default shall be nometoo.

onehop When responding to a message that was originally sent to sev? eral recipients, the other recipient addresses are normally forced to be relative to the originating author's machine for the response. This flag disables alteration of the recipi? ents' addresses, improving efficiency in a network where all machines can send directly to all other machines (that is, one hop away). The default shall be noonehop.

outfolder Cause the files used to record outgoing messages to be lo? cated in the directory specified by the folder variable un? less the pathname is absolute. The default shall be noout? folder. See the record variable.

page Insert a <form-feed> after each message sent through the pipe created by the pipe command. The default shall be nopage.

prompt=string

Set the command-mode prompt to string. If string is null or if noprompt is set, no prompting shall occur. The default shall be to prompt with the string "? ".

quiet Refrain from writing the opening message and version when en? tering mailx. The default shall be noquiet.

record=file

Record all outgoing mail in the file with the pathname file. The default shall be norecord. See also outfolder above.

save Enable saving of messages in the dead-letter file on inter? rupt or delivery error. See the variable DEAD for the loca? tion of the dead-letter file. The default shall be save.

screen=number

Set the number of lines in a screenful of headers for the headers and z commands. If screen is not specified, a value based on the terminal type identified by the TERM environment

variable, the window size, the baud rate, or some combination of these shall be used.

sendwait Wait for the background mailer to finish before returning. The default shall be nosendwait.

showto When the sender of the message was the user who is invoking mailx, write the information from the To: line instead of the From: line in the header summary. The default shall be noshowto.

sign=string

Set the variable inserted into the text of a message when the ~a command escape is given. The default shall be nosign. The character sequences '\t' and '\n' shall be recognized in the variable as <tab> and <newline> characters, respectively. (See also ~i in Command Escapes in mailx.)

Sign=string

Set the variable inserted into the text of a message when the ~A command escape is given. The default shall be noSign. The character sequences '\t' and '\n' shall be recognized in the variable as <tab> and <newline> characters, respectively.

toplines=number

Set the number of lines of the message to write with the top command. The default shall be 5.

Commands in mailx

The following mailx commands shall be provided. In the following list, header refers to lines from the message header, as shown in the OUTPUT FILES section. Header-line refers to lines within the header that be? gin with one or more non-white-space characters, immediately followed by a <colon> and white space and continuing until the next line begin? ning with a non-white-space character or an empty line. Header-field refers to the portion of a header line prior to the first <colon> in that line.

For each of the commands listed below, the command can be entered as the abbreviation (those characters in the Synopsis command word preced?

ing the '['), the full command (all characters shown for the command word, omitting the '[' and ']'), or any truncation of the full command down to the abbreviation. For example, the exit command (shown as ex[it] in the Synopsis) can be entered as ex, exi, or exit.

The arguments to commands can be quoted, using the following methods:

* An argument can be enclosed between paired double-quotes ("") or single-quotes ('); any white space, shell word expansion, or <backslash> characters within the quotes shall be treated literally as part of the argument. A double-quote shall be treated literally within single-quotes and vice versa. These special properties of the <quotation-mark> characters shall occur only when they are paired at the beginning and end of the argument.
* A <backslash> outside of the enclosing quotes shall be discarded and the following character treated literally as part of the argu? ment.
* An unquoted <backslash> at the end of a command line shall be dis? carded and the next line shall continue the command.

Filenames, where expected, shall be subjected to the following trans? formations, in sequence:

* If the filename begins with an unquoted <plus-sign>, and the folder variable is defined (see the folder variable), the <plus-sign> shall be replaced by the value of the folder variable followed by a <slash>. If the folder variable is unset or is set to null, the filename shall be unchanged.
* Shell word expansions shall be applied to the filename (see Section 2.6, Word Expansions). If more than a single pathname results from this expansion and the command is expecting one file, the effects are unspecified.

Declare Aliases

Synopsis:

a[lias] [alias [address...]]

g[roup] [alias [address...]]

Add the given addresses to the alias specified by alias. The names

shall be substituted when alias is used as a recipient address speci? fied by the user in an outgoing message (that is, other recipients ad? dressed indirectly through the reply command shall not be substituted in this manner). Mail address alias substitution shall apply only when the alias string is used as a full address; for example, when hlj is an alias, hlj@posix.com does not trigger the alias substitution. If no ar? guments are given, write a listing of the current aliases to standard output. If only an alias argument is given, write a listing of the specified alias to standard output. These listings need not reflect the same order of addresses that were entered.

Declare Alternatives

Synopsis:

alt[ernates] name...

(See also the metoo variable.) Declare a list of alternative names for the user's login. When responding to a message, these names shall be removed from the list of recipients for the response. The comparison of names shall be in a case-insensitive manner. With no arguments, alter? nates shall write the current list of alternative names.

Change Current Directory

Synopsis:

cd [directory]

ch[dir] [directory]

Change directory. If directory is not specified, the contents of HOME shall be used.

Copy Messages

Synopsis:

c[opy] [file]

c[opy] [msglist] file

C[opy] [msglist]

Copy messages to the file named by the pathname file without marking the messages as saved. Otherwise, it shall be equivalent to the save command.

In the capitalized form, save the specified messages in a file whose

name is derived from the author of the message to be saved, without marking the messages as saved. Otherwise, it shall be equivalent to the Save command.

Delete Messages

Synopsis:

d[elete] [msglist]

Mark messages for deletion from the mailbox. The deletions shall not occur until mailx quits (see the quit command) or changes mailboxes (see the folder command). If autoprint is set and there are messages remaining after the delete command, the current message shall be writ? ten as described for the print command (see the print command); other? wise, the mailx prompt shall be written.

Discard Header Fields

Synopsis:

di[scard] [header-field...]

ig[nore] [header-field...]

Suppress the specified header fields when writing messages. Specified header-fields shall be added to the list of suppressed header fields. Examples of header fields to ignore are status and cc. The fields shall be included when the message is saved. The Print and Type com? mands shall override this command. The comparison of header fields shall be in a case-insensitive manner. If no arguments are specified, write a list of the currently suppressed header fields to standard out? put; the listing need not reflect the same order of header fields that were entered.

If both retain and discard commands are given, discard commands shall be ignored.

Delete Messages and Display

Synopsis:

dp [msglist]

dt [msglist]

Delete the specified messages as described for the delete command, ex? cept that the autoprint variable shall have no effect, and the current

message shall be written only if it was set to a message after the last message deleted by the command. Otherwise, an informational message to the effect that there are no further messages in the mailbox shall be written, followed by the mailx prompt.

Echo a String

Synopsis:

ec[ho] string ...

Echo the given strings, equivalent to the shell echo utility.

Edit Messages

Synopsis:

e[dit] [msglist]

Edit the given messages. The messages shall be placed in a temporary file and the utility named by the EDITOR variable is invoked to edit each file in sequence. The default EDITOR is unspecified.

The edit command does not modify the contents of those messages in the mailbox.

Exit

Synopsis:

ex[it]

x[it]

Exit from mailx without changing the mailbox. No messages shall be saved in the mbox (see also quit).

Change Folder

Synopsis:

fi[le] [file]

fold[er] [file]

Quit (see the quit command) from the current file of messages and read in the file named by the pathname file. If no argument is given, the name and status of the current mailbox shall be written.

Several unquoted special characters shall be recognized when used as file names, with the following substitutions:

% The system mailbox for the invoking user.

%user The system mailbox for user.

The previous file.

& The current mbox.

+file The named file in the folder directory. (See the folder vari?

able.)

The default file shall be the current mailbox.

Display List of Folders

Synopsis:

folders

Write the names of the files in the directory set by the folder vari?

able. The command specified by the LISTER environment variable shall be

used (see the ENVIRONMENT VARIABLES section).

Follow Up Specified Messages

Synopsis:

fo[llowup] [message]

F[ollowup] [msglist]

In the lowercase form, respond to a message, recording the response in

a file whose name is derived from the author of the message. See also

the save and copy commands and outfolder.

In the capitalized form, respond to the first message in the msglist,

sending the message to the author of each message in the msglist. The

subject line shall be taken from the first message and the response

shall be recorded in a file whose name is derived from the author of

the first message. See also the Save and Copy commands and outfolder.

Both forms shall override the record variable, if set.

Display Header Summary for Specified Messages

Synopsis:

f[rom] [msglist]

Write the header summary for the specified messages.

Display Header Summary

Synopsis:

h[eaders] [message]

Write the page of headers that includes the message specified. If the

message argument is not specified, the current message shall not

change. However, if the message argument is specified, the current message shall become the message that appears at the top of the page of headers that includes the message specified. The screen variable sets the number of headers per page. See also the z command.

Help

Synopsis:

hel[p]

?

Write a summary of commands.

Hold Messages

Synopsis:

ho[ld] [msglist]

pre[serve] [msglist]

Mark the messages in msglist to be retained in the mailbox when mailx terminates. This shall override any commands that might previously have marked the messages to be deleted. During the current invocation of mailx, only the delete, dp, or dt commands shall remove the preserve marking of a message.

Execute Commands Conditionally

Synopsis:

i[f] s|r

mail-commands

el[se]

mail-commands

en[dif]

Execute commands conditionally, where if s executes the following mail-commands, up to an else or endif, if the program is in Send Mode, and if r shall cause the mail-commands to be executed only in Receive Mode.

List Available Commands

Synopsis:

l[ist]

Write a list of all commands available. No explanation shall be given.

Mail a Message

Synopsis:

m[ail] address...

Mail a message to the specified addresses or aliases.

Direct Messages to mbox

Synopsis:

mb[ox] [msglist]

Arrange for the given messages to end up in the mbox save file when mailx terminates normally. See MBOX. See also the exit and quit com? mands.

Process Next Specified Message

Synopsis:

n[ext] [message]

If the current message has not been written (for example, by the print command) since mailx started or since any other message was the current message, behave as if the print command was entered. Otherwise, if there is an undeleted message after the current message, make it the current message and behave as if the print command was entered. Other? wise, an informational message to the effect that there are no further messages in the mailbox shall be written, followed by the mailx prompt. Should the current message location be the result of an immediately preceding hold, mbox, preserve, or touch command, next will act as if the current message has already been written.

Pipe Message

Synopsis:

pi[pe] [[msglist] command]

| [[msglist] command]

Pipe the messages through the given command by invoking the command in? terpreter specified by SHELL with two arguments: -c and command. (See also sh -c.) The application shall ensure that the command is given as a single argument. Quoting, described previously, can be used to accom? plish this. If no arguments are given, the current message shall be piped through the command specified by the value of the cmd variable. If the page variable is set, a <form-feed> shall be inserted after each

message.

Display Message with Headers

Synopsis:

P[rint] [msglist]

T[ype] [msglist]

Write the specified messages, including all header lines, to standard output. Override suppression of lines by the discard, ignore, and re? tain commands. If crt is set, the messages longer than the number of lines specified by the crt variable shall be paged through the command specified by the PAGER environment variable.

Display Message

Synopsis:

p[rint] [msglist]

t[ype] [msglist]

Write the specified messages to standard output. If crt is set, the messages longer than the number of lines specified by the crt variable shall be paged through the command specified by the PAGER environment variable.

Quit

Synopsis:

q[uit]

end-of-file

Terminate mailx, storing messages that were read in mbox (if the cur? rent mailbox is the system mailbox and unless hold is set), deleting messages that have been explicitly saved (unless keepsave is set), dis? carding messages that have been deleted, and saving all remaining mes? sages in the mailbox.

Reply to a Message List

Synopsis:

R[eply] [msglist]

R[espond] [msglist]

Mail a reply message to the sender of each message in the msglist. The subject line shall be formed by concatenating Re:<space> (unless it al?

ready begins with that string) and the subject from the first message.

If record is set to a filename, the response shall be saved at the end of that file.

See also the flipr variable.

Reply to a Message

Synopsis:

r[eply] [message]

r[espond] [message]

Mail a reply message to all recipients included in the header of the message. The subject line shall be formed by concatenating Re:<space> (unless it already begins with that string) and the subject from the message. If record is set to a filename, the response shall be saved at the end of that file.

See also the flipr variable.

Retain Header Fields

Synopsis:

ret[ain] [header-field...]

Retain the specified header fields when writing messages. This command shall override all discard and ignore commands. The comparison of header fields shall be in a case-insensitive manner. If no arguments are specified, write a list of the currently retained header fields to standard output; the listing need not reflect the same order of header fields that were entered.

Save Messages

Synopsis:

s[ave] [file]

s[ave] [msglist] file

S[ave] [msglist]

Save the specified messages in the file named by the pathname file, or the mbox if the file argument is omitted. The file shall be created if it does not exist; otherwise, the messages shall be appended to the file. The message shall be put in the state saved, and shall behave as specified in the description of the saved state when the current mail?

box is exited by the quit or file command.

In the capitalized form, save the specified messages in a file whose name is derived from the author of the first message. The name of the file shall be taken to be the author's name with all network addressing stripped off. See also the Copy, followup, and Followup commands and outfolder variable.

Set Variables

Synopsis:

se[t] [name[=[string]] ...] [name=number ...] [noname ...]

Define one or more variables called name. The variable can be given a null, string, or numeric value. Quoting and <backslash>-escapes can oc? cur anywhere in string, as described previously, as if the string por? tion of the argument were the entire argument. The forms name and name= shall be equivalent to name="" for variables that take string values.

The set command without arguments shall write a list of all defined variables and their values. The no name form shall be equivalent to un? set name.

Invoke a Shell

Synopsis:

sh[ell]

Invoke an interactive command interpreter (see also SHELL).

Display Message Size

Synopsis:

si[ze] [msglist]

Write the size in bytes of each of the specified messages.

Read mailx Commands From a File

Synopsis:

so[urce] file

Read and execute commands from the file named by the pathname file and return to command mode.

Display Beginning of Messages

Synopsis:

to[p] [msglist]

Write the top few lines of each of the specified messages. If the toplines variable is set, it is taken as the number of lines to write. The default shall be 5.

Touch Messages

Synopsis:

tou[ch] [msglist]

Touch the specified messages. If any message in msglist is not specifically deleted nor saved in a file, it shall be placed in the mbox upon normal termination. See exit and quit.

Delete Aliases

Synopsis:

una[lias] [alias]...

Delete the specified alias names. If a specified alias does not exist, the results are unspecified.

Undelete Messages

Synopsis:

u[ndelete] [msglist]

Change the state of the specified messages from deleted to read. If autoprint is set, the last message of those restored shall be written. If msglist is not specified, the message shall be selected as follows:

* If there are any deleted messages that follow the current message, the first of these shall be chosen.
* Otherwise, the last deleted message that also precedes the current message shall be chosen.

Unset Variables

Synopsis:

uns[et] name...

Cause the specified variables to be erased.

Edit Message with Full-Screen Editor

Synopsis:

v[isual] [msglist]

Edit the given messages with a screen editor. Each message shall be placed in a temporary file, and the utility named by the VISUAL vari?

able shall be invoked to edit each file in sequence. The default editor shall be vi.

The visual command does not modify the contents of those messages in the mailbox.

Write Messages to a File

Synopsis:

w[rite] [msglist] file

Write the given messages to the file specified by the pathname file, minus the message header. Otherwise, it shall be equivalent to the save command.

Scroll Header Display

Synopsis:

z[+|-]

Scroll the header display forward (if '+' is specified or if no option is specified) or backward (if '-' is specified) one screenful. The num? ber of headers written shall be set by the screen variable.

Invoke Shell Command

Synopsis:

!command

Invoke the command interpreter specified by SHELL with two arguments: -c and command. (See also sh -c.) If the bang variable is set, each unescaped occurrence of '!' in command shall be replaced with the com? mand executed by the previous ! command or ~! command escape.

Null Command

Synopsis:

comment

This null command (comment) shall be ignored by mailx.

Display Current Message Number

Synopsis:

=

Write the current message number.

Command Escapes in mailx

The following commands can be entered only from input mode, by begin?

ning a line with the escape character (by default, <tilde> ('~')). See the escape variable description for changing this special character.

The format for the commands shall be:

<escape-character><command-char><separator>[<arguments>]

where the <separator> can be zero or more <blank> characters.

In the following descriptions, the application shall ensure that the argument command (but not mailx-command) is a shell command string. Any string acceptable to the command interpreter specified by the SHELL variable when it is invoked as SHELL -c command_string shall be valid. The command can be presented as multiple arguments (that is, quoting is not required).

Command escapes that are listed with msglist or mailx-command arguments are invalid in Send Mode and produce unspecified results.

~! command

Invoke the command interpreter specified by SHELL with two arguments: -c and command; and then return to input mode. If the bang variable is set, each unescaped occurrence of '!' in command shall be replaced with the command executed by the previous ! command or ~! command escape.

~. Simulate end-of-file (terminate message input).

~: mailx-command, ~_ mailx-command

Perform the command-level request.

~? Write a summary of command escapes.

~A This shall be equivalent to ~i Sign.

~a This shall be equivalent to ~i sign.

~b name...

Add the names to the blind carbon copy (Bcc) list.

~c name...

Add the names to the carbon copy (Cc) list.

~d Read in the dead-letter file. See DEAD for a description of this file.

~e Invoke the editor, as specified by the EDITOR environment variable, on the partial message.

~f [msglist]

Forward the specified messages. The specified messages shall be inserted into the current message without alteration. This command escape also shall insert message headers into the message with field selection affected by the discard, ignore, and retain commands.

~F [msglist]

This shall be the equivalent of the ~f command escape, except that all headers shall be included in the message, regardless of previous discard, ignore, and retain commands.

~h If standard input is a terminal, prompt for a Subject line and the To, Cc, and Bcc lists. Other implementation-defined headers may also be presented for editing. If the field is written with an initial value, it can be edited as if it had just been typed.

~i string Insert the value of the named variable, followed by a <new? line>, into the text of the message. If the string is unset or null, the message shall not be changed.

~m [msglist]

Insert the specified messages into the message, prefixing non-empty lines with the string in the indentprefix variable. This command escape also shall insert message headers into the message, with field selection affected by the discard, ignore, and retain commands.

~M [msglist]

This shall be the equivalent of the ~m command escape, except that all headers shall be included in the message, regardless of previous discard, ignore, and retain commands.

~p Write the message being entered. If the message is longer than crt lines (see Internal Variables in mailx), the output shall be paginated as described for the PAGER variable.

~q Quit (see the quit command) from input mode by simulating an interrupt. If the body of the message is not empty, the par?

tial message shall be saved in the dead-letter file. See DEAD for a description of this file.

~r file, ~< file, ~r !command, ~< !command

Read in the file specified by the pathname file. If the argument begins with an <exclamation-mark> ('!'), the rest of the string shall be taken as an arbitrary system command; the command interpreter specified by SHELL shall be invoked with two arguments: -c and command. The standard output of command shall be inserted into the message.

~s string Set the subject line to string.

~t name...

Add the given names to the To list.

~v Invoke the full-screen editor, as specified by the VISUAL environment variable, on the partial message.

~w file Write the partial message, without the header, onto the file named by the pathname file. The file shall be created or the message shall be appended to it if the file exists.

~x Exit as with ~q, except the message shall not be saved in the dead-letter file.

~| command

Pipe the body of the message through the given command by invoking the command interpreter specified by SHELL with two arguments: -c and command. If the command returns a successful exit status, the standard output of the command shall replace the message. Otherwise, the message shall remain unchanged. If the command fails, an error message giving the exit status shall be written.

EXIT STATUS

When the -e option is specified, the following exit values are returned:

0 Mail was found.

>0 Mail was not found or an error occurred.

Otherwise, the following exit values are returned:

0 Successful completion; note that this status implies that all messages were sent, but it gives no assurances that any of them were actually delivered.

>0 An error occurred.

CONSEQUENCES OF ERRORS

When in input mode (Receive Mode) or Send Mode:

* If an error is encountered processing an input line beginning with a <tilde> ('~') character, (see Command Escapes in mailx), a diagnostic message shall be written to standard error, and the message being composed may be modified, but this condition shall not prevent the message from being sent.
* Other errors shall prevent the sending of the message.

When in command mode:

* Default.

The following sections are informative.

APPLICATION USAGE

Delivery of messages to remote systems requires the existence of communication paths to such systems. These need not exist.

Input lines are limited to {LINE_MAX} bytes, but mailers between systems may impose more severe line-length restrictions. This volume of POSIX.1?2017 does not place any restrictions on the length of messages handled by mailx, and for delivery of local messages the only limitations should be the normal problems of available disk space for the target mail file. When sending messages to external machines, applications are advised to limit messages to less than 100000 bytes because some mail gateways impose message-length restrictions.

The format of the system mailbox is intentionally unspecified. Not all systems implement system mailboxes as flat files, particularly with the advent of multimedia mail messages. Some system mailboxes may be multiple files, others records in a database. The internal format of the messages themselves is specified with the historical format from Version 7, but only after the messages have been saved in some file other than the system mailbox. This was done so that many historical applica?

tions expecting text-file mailboxes are not broken.

Some new formats for messages can be expected in the future, probably including binary data, bit maps, and various multimedia objects. As described here, mailx is not prohibited from handling such messages, but it must store them as text files in secondary mailboxes (unless some extension, such as a variable or command line option, is used to change the stored format). Its method of doing so is implementation-defined and might include translating the data into text file-compatible or readable form or omitting certain portions of the message from the stored output.

The discard and ignore commands are not inverses of the retain command. The retain command discards all header-fields except those explicitly retained. The discard command keeps all header-fields except those explicitly discarded. If headers exist on the retained header list, discard and ignore commands are ignored.

EXAMPLES

None.

RATIONALE

The standard developers felt strongly that a method for applications to send messages to specific users was necessary. The obvious example is a batch utility, running non-interactively, that wishes to communicate errors or results to a user. However, the actual format, delivery mechanism, and method of reading the message are clearly beyond the scope of this volume of POSIX.1?2017.

The intent of this command is to provide a simple, portable interface for sending messages non-interactively. It merely defines a ``front-end'' to the historical mail system. It is suggested that implementations explicitly denote the sender and recipient in the body of the delivered message. Further specification of formats for either the message envelope or the message itself were deliberately not made, as the industry is in the midst of changing from the current standards to a more internationalized standard and it is probably incorrect, at this time, to require either one.

Implementations are encouraged to conform to the various delivery mech?
anisms described in the CCITT X.400 standards or to the equivalent In?
ternet standards, described in Internet Request for Comment (RFC) docu?
ments RFC 819, RFC 920, RFC 921, RFC 1123, and RFC 5322 (which suc?
ceeded RFC 822).

Many historical systems modified each body line that started with From
by prefixing the 'F' with '>'. It is unnecessary, but allowed, to do
that when the string does not follow a blank line because it cannot be
confused with the next header.

The edit and visual commands merely edit the specified messages in a
temporary file. They do not modify the contents of those messages in
the mailbox; such a capability could be added as an extension, such as
by using different command names.

The restriction on a subject line being {LINE_MAX}-10 bytes is based on
the historical format that consumes 10 bytes for Subject: and the
trailing <newline>. Many historical mailers that a message may en?
counter on other systems are not able to handle lines that long, how?
ever.

Like the utilities logger and lp, mailx admittedly is difficult to
test. This was not deemed sufficient justification to exclude this
utility from this volume of POSIX.1?2017. It is also arguable that it
is, in fact, testable, but that the tests themselves are not portable.

When mailx is being used by an application that wishes to receive the
results as if none of the User Portability Utilities option features
were supported, the DEAD environment variable must be set to /dev/null.
Otherwise, it may be subject to the file creations described in mailx
ASYNCHRONOUS EVENTS. Similarly, if the MAILRC environment variable is
not set to /dev/null, historical versions of mailx and Mail read ini?
tialization commands from a file before processing begins. Since the
initialization that a user specifies could alter the contents of mes?
sages an application is trying to send, such applications must set
MAILRC to /dev/null.

The description of LC_TIME uses ``may affect'' because many historical

implementations do not or cannot manipulate the date and time strings in the incoming mail headers. Some headers found in incoming mail do not have enough information to determine the timezone in which the mail originated, and, therefore, mailx cannot convert the date and time strings into the internal form that then is parsed by routines like strftime() that can take LC_TIME settings into account. Changing all these times to a user-specified format is allowed, but not required.

The paginator selected when PAGER is null or unset is partially unspec? ified to allow the System V historical practice of using pg as the de? fault. Bypassing the pagination function, such as by declaring that cat is the paginator, would not meet with the intended meaning of this de? scription. However, any ``portable user'' would have to set PAGER ex? plicitly to get his or her preferred paginator on all systems. The pag? inator choice was made partially unspecified, unlike the VISUAL editor choice (mandated to be vi) because most historical pagers follow a com? mon theme of user input, whereas editors differ dramatically.

Options to specify addresses as cc (carbon copy) or bcc (blind carbon copy) were considered to be format details and were omitted.

A zero exit status implies that all messages were sent, but it gives no assurances that any of them were actually delivered. The reliability of the delivery mechanism is unspecified and is an appropriate market? ing distinction between systems.

In order to conform to the Utility Syntax Guidelines, a solution was required to the optional file option-argument to -f. By making file an operand, the guidelines are satisfied and users remain portable. How? ever, it does force implementations to support usage such as:

```
mailx -fin mymail.box
```

The no name method of unsetting variables is not present in all histor? ical systems, but it is in System V and provides a logical set of com? mands corresponding to the format of the display of options from the mailx set command without arguments.

The ask and asksub variables are the names selected by BSD and System V, respectively, for the same feature. They are synonyms in this volume

of POSIX.1?2017.

The mailx echo command was not documented in the BSD version and has been omitted here because it is not obviously useful for interactive users.

The default prompt on the System V mailx is a <question-mark>, on BSD Mail an <ampersand>. Since this volume of POSIX.1?2017 chose the mailx name, it kept the System V default, assuming that BSD users would not have difficulty with this minor incompatibility (that they can over? ride).

The meanings of r and R are reversed between System V mailx and SunOS Mail. Once again, since this volume of POSIX.1?2017 chose the mailx name, it kept the System V default, but allows the SunOS user to achieve the desired results using flipr, an internal variable in System V mailx, although it has not been documented in the SVID.

The indentprefix variable, the retain and unalias commands, and the ~F and ~M command escapes were adopted from 4.3 BSD Mail.

The version command was not included because no sufficiently general specification of the version information could be devised that would still be useful to a portable user. This command name should be used by suppliers who wish to provide version information about the mailx com? mand.

The ``implementation-specific (unspecified) system start-up file'' his? torically has been named /etc/mailx.rc, but this specific name and lo? cation are not required.

The intent of the wording for the next command is that if any command has already displayed the current message it should display a following message, but, otherwise, it should display the current message. Con? sider the command sequence:

```
next 3
delete 3
next
```

where the autoprint option was not set. The normative text specifies that the second next command should display a message following the

third message, because even though the current message has not been displayed since it was set by the delete command, it has been displayed since the current message was anything other than message number 3. This does not always match historical practice in some implementations, where the command file address followed by next (or the default com? mand) would skip the message for which the user had searched.

FUTURE DIRECTIONS

None.

SEE ALSO

Chapter 2, Shell Command Language, ed, ls, more, vi

The Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guidelines

COPYRIGHT